自动化测试综合实战

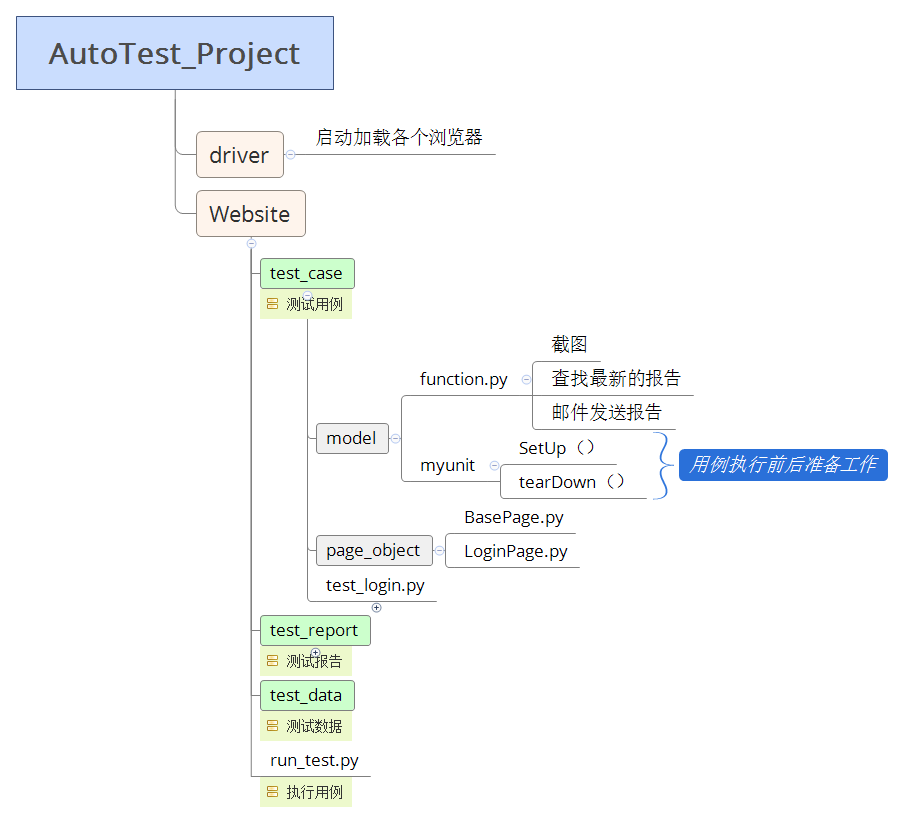
**项目背景**

在 <http://localhost/news/> 新闻子页面进行登录测试。

**功能实现**

* 自动运行用例
* 自动生成测试报告
* 自动断言与截图
* 自动将最新测试报告发送到指定邮箱
* PageObject+Unittest

**项目架构**

****

浏览器driver定义

**from** selenium **import** webdriver

*#启动浏览器驱动*

**def** **browser**():

driver = webdriver.Firefox()

*# driver = webdriver.Chrome()*

*# driver = webdriver.Ie()*

*# driver=webdriver.PhantomJS()*

*# driver.get("http://www.baidu.com")*

**return** driver

*#调试运行*

**if** \_\_name\_\_ == '\_\_main\_\_':

browser()

用例运行前后的环境准备工作

**import** unittest

**from** driver **import** \*

**class** **StartEnd**(unittest.TestCase):

**def** **setUp**(self):

self.driver=browser()

self.driver.implicitly\_wait(10)

self.driver.maximize\_window()

**def** **tearDown**(self):

self.driver.quit()

工具方法模块（截图，查找最新报告、邮件发送）

**from** selenium **import** webdriver

**import** os

**import** smtplib

**from** email.mime.text **import** MIMEText

**from** email.header **import** Header

*#截图方法*

**def** **inser\_img**(driver,filename):

*#获取当前模块所在路径*

func\_path=os.path.dirname(\_\_file\_\_)

*# print("func\_path is %s" %func\_path)*

*#获取test\_case目录*

base\_dir=os.path.dirname(func\_path)

*# print("base\_dir is %s" %base\_dir)*

*#将路径转化为字符串*

base\_dir=str(base\_dir)

*#对路径的字符串进行替换*

base\_dir=base\_dir.replace('\\','/')

*# print(base\_dir)*

*#获取项目文件的根目录路径*

base=base\_dir.split('/Website')[0]

*# print(base)*

*#指定截图存放路径*

filepath=base+'/Website/test\_report/screenshot/'+filename

*# print(filepath)*

driver.get\_screenshot\_as\_file(filepath)

*#查找最新的测试报告*

**def** **latest\_report**(report\_dir):

lists = os.listdir(report\_dir)

*# print(lists)*

lists.sort(key=**lambda** fn: os.path.getatime(report\_dir + '\\' + fn))

*# print("the latest report is " + lists[-1])*

file = os.path.join(report\_dir, lists[-1])

*# print(file)*

**return** file

*#将测试报告发送到邮件*

**def** **send\_mail**(latest\_report):

f=open(latest\_report,'rb')

mail\_content=f.read()

f.close()

smtpserver = 'smtp.163.com'

user = 'yuexiaolu2015@163.com'

password = '...'

sender = 'yuexiaolu2015@163.com'

receives = ['yuexiaolu2015@126.com', 'yuexiaolu2015@sina.com']

subject = 'Web Selenium 自动化测试报告'

msg = MIMEText(mail\_content, 'html', 'utf-8')

msg['Subject'] = Header(subject, 'utf-8')

msg['From'] = sender

msg['To'] = ','.join(receives)

smtp = smtplib.SMTP\_SSL(smtpserver, 465)

smtp.helo(smtpserver)

smtp.ehlo(smtpserver)

smtp.login(user, password)

print("Start send email...")

smtp.sendmail(sender, receives, msg.as\_string())

smtp.quit()

print("Send email end!")

**if** \_\_name\_\_ == '\_\_main\_\_':

driver=webdriver.Firefox()

driver.get("http://www.sogou.com")

inser\_img(driver,"sogou.png")

driver.quit()

#### Pageobject页面对象封装

[BasePage.py](http://BasePage.py) —— 基础页面类

**from** time **import** sleep

**class** **Page**():

**def** **\_\_init\_\_**(self,driver):

self.driver=driver

self.base\_url="http://localhost"

self.timeout=20

**def** **\_open**(self,url):

url\_=self.base\_url+url

print('Test page is:%s' %url\_)

self.driver.maximize\_window()

self.driver.get(url\_)

sleep(2)

**assert** self.driver.current\_url == url\_, 'Did ont land on %s' % url\_

**def** **open**(self):

self.\_open(self.url)

**def** **find\_element**(self,\*loc):

**return** self.driver.find\_element(\*loc)

[LoginPage.py](http://LoginPage.py) —— 新闻登录页面

**from** PageBase **import** \*

**from** selenium **import** webdriver

**from** selenium.webdriver.common.by **import** By

**class** **LoginPage**(Page):

'''新闻登录页面'''

url = '/news/'

*# 定位器——对相关元素进行定位*

username\_loc = (By.NAME, 'username')

password\_loc = (By.NAME, 'password')

submit\_loc = (By.NAME, 'Submit')

**def** **type\_username**(self, username):

self.find\_element(\*self.username\_loc).clear()

self.find\_element(\*self.username\_loc).send\_keys(username)

**def** **type\_password**(self, password):

self.find\_element(\*self.password\_loc).clear()

self.find\_element(\*self.password\_loc).send\_keys(password)

**def** **type\_submit**(self):

self.find\_element(\*self.submit\_loc).click()

**def** **Login\_action**(self,username,password):

self.open()

self.type\_username(username)

self.type\_password(password)

self.type\_submit()

LoginPass\_loc = (By.LINK\_TEXT, '我的空间')

loginFail\_loc = (By.NAME, 'username')

**def** **type\_loginPass\_hint**(self):

**return** self.find\_element(\*self.LoginPass\_loc).text

**def** **type\_loginFail\_hint**(self):

**return** self.find\_element(\*self.loginFail\_loc).text

test\_login.py ——unittest组织测试用例

* 用户名密码正确点击登录
* 用户名正确，密码错误点击登录
* 用户名和密码为空点击登录

**import** unittest

**from** model **import** function,myunit

**from** page\_object.LoginPage **import** \*

**from** time **import** sleep

**class** **LoginTest**(myunit.StartEnd):

*# @unittest.skip('skip this case')*

**def** **test\_login1\_normal**(self):

'''username password is normal'''

print("test\_login1\_normal is start run...")

po=LoginPage(self.driver)

po.Login\_action('51zxw',123456)

sleep(3)

*#断言与截屏*

self.assertEqual(po.type\_loginPass\_hint(),'我的空间')

function.insert\_img(self.driver,"51zxw\_login1\_normal.jpg")

print("test\_login1\_normal is test end!")

**def** **test\_login2\_PasswdError**(self):

'''username is ok,passwd is error!'''

print("test\_login2\_PasswdError is start run...")

po=LoginPage(self.driver)

po.Login\_action("51zxw",12342)

sleep(2)

self.assertEqual(po.type\_loginFail\_hint(),'')

function.insert\_img(self.driver,"51zxw\_login2\_fail.jpg")

print("test\_login2\_PasswdError is test end!")

*#*

*# @unittest.skip*

**def** **test\_login3\_empty**(self):

'''username password is empty'''

print("test\_login3\_empty is start run...")

po=LoginPage(self.driver)

po.Login\_action('','')

sleep(2)

*#断言与截屏*

self.assertEqual(po.type\_loginFail\_hint(),'')

function.insert\_img(self.driver,"51zxw\_login3\_empty.jpg")

print("test\_login3\_empty is test end!")

**if** \_\_name\_\_ == '\_\_main\_\_':

unittest.main()

run\_test.py——执行测试用例

import unittest

from function import \*

from BSTestRunner import BSTestRunner

import **time**

report\_dir = './test\_report'

test\_dir = './test\_case'

**print**("start run testcase...")

discover = unittest.defaultTestLoader.discover(test\_dir, pattern="test\_login.py")

now = **time**.strftime("%Y-%m-%d %H\_%M\_%S")

report\_name = report\_dir + '/' + now + 'result.html'

**print**("start write report...")

*# 运行前记得把BSTestRunner.py 120行‘unicode’ 换成‘str’*

with **open**(report\_name, 'wb') as f:

runner = BSTestRunner(stream=f, title="Test Report", description="localhost login test")

runner.run(discover)

f.**close**()

**print**("find latest report...")

*# 查找最新的测试报告*

latest\_report = latest\_report(report\_dir)

*# 邮件发送报告*

**print**("send email report...")

send\_mail(latest\_report)

**print**("test end!")

### 浏览器内核

Webkit：目前最主流的浏览器内核，webkit是苹果公司开源的浏览器内核，其前身是KHTML。基于Webkit的浏览器很多，比如Safari，Chrome，Opera

Gecko：是Firefox浏览器的内核

Trident：是IE浏览器的内核

Blink：是webkit的一个分支版本，由google开发

### 无头浏览器

无头浏览器即headless browser，是一种没有界面的浏览器。既然是浏览器那么浏览器该有的东西它都应该有，只是看不到界面而已。

### PhantomJS

PhantomJS is a headless WebKit scriptable with a JavaScript API. It has fast and native support for various web standards: DOM handling, CSS selector, JSON, Canvas, and SVG.

PhantomJS可以说是目前使用最为广泛，也是最被认可的无头浏览器。由于采用的是Webkit内核，因此其和目前的Safari，Chrome等浏览器兼容性十分好。

### 为什么要使用PhantomJS？

PhantomJS 是一个无界面, 基于Webkit 的javascript 引擎. 一般来说我们的自动化脚本是须要运行在服务器上的, 往往这个时候系统并没有图形界面(如liunx服务器), 或者配置太低跑个浏览器实在是浪费. 而不需要图形界面的 PhantomJS 可谓是我们的不二之选.

### PhantomJS安装配置

下载地址：<http://phantomjs.org/download.html>

下载完成后将phantomjs.exe文件放置Python安装目录即可。

运行使用

driver=webdriver.**PhantomJS**()

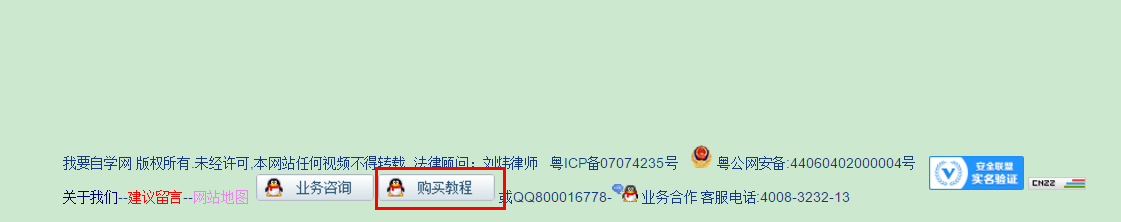
自学网App下载：

<http://www.51zxw.net/zt/aboutapp/downapp.html>



光盘购买：

点击网站底部“购买教程”按钮即可联系工作人员



参考文档

<http://www.cnblogs.com/diaosicai/p/6370321.html>

<http://blog.csdn.net/xtuhcy/article/details/52814210?locationNum=5>